**Exercise 4: Functions**

**Scenario 1:**

**CREATE OR REPLACE FUNCTION CalculateAge (**

**p\_dob DATE**

**) RETURN NUMBER IS**

**v\_age NUMBER;**

**BEGIN**

**v\_age := TRUNC(MONTHS\_BETWEEN(SYSDATE, p\_dob) / 12);**

**RETURN v\_age;**

**END CalculateAge;**

**/**

**Scenario 2:**

**CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (**

**p\_loan\_amount NUMBER,**

**p\_interest\_rate NUMBER,**

**p\_duration\_years NUMBER**

**) RETURN NUMBER IS**

**v\_monthly\_rate NUMBER;**

**v\_num\_payments NUMBER;**

**v\_monthly\_installment NUMBER;**

**BEGIN**

**v\_monthly\_rate := p\_interest\_rate / 12 / 100;**

**v\_num\_payments := p\_duration\_years \* 12;**

**v\_monthly\_installment := p\_loan\_amount \* v\_monthly\_rate / (1 - POWER(1 + v\_monthly\_rate, -v\_num\_payments));**

**RETURN v\_monthly\_installment;**

**END CalculateMonthlyInstallment;**

**/**

**Scenario 3:**

**CREATE OR REPLACE FUNCTION HasSufficientBalance (**

**p\_account\_id NUMBER,**

**p\_amount NUMBER**

**) RETURN BOOLEAN IS**

**v\_balance NUMBER;**

**BEGIN**

**SELECT balance INTO v\_balance FROM accounts WHERE account\_id = p\_account\_id;**

**IF v\_balance >= p\_amount THEN**

**RETURN TRUE;**

**ELSE**

**RETURN FALSE;**

**END IF;**

**EXCEPTION**

**WHEN NO\_DATA\_FOUND THEN**

**RETURN FALSE;**

**END HasSufficientBalance;**

**/**